Model for the Exchange of Descriptive Information (XDI)

Lee Neitzel
June 1997

Part 1:
Background

What is Descriptive Information?

Descriptive Information - identifies, defines, categorizes, and relates the components used to construct system. and their interrelationships

Descriptive Information - describes the configuration of a system using component descriptions.

Descriptive Information - typically stored in a custom format, e.g. database, file system, spreadsheet.

What is the Exchange Of Descriptive Information?

The transfer of system specification data and/or configuration data from one data base, file system, spreadsheet, etc. to another.

What is Necessary to Exchange Descriptive Information?

Information Identifier:

Each piece of information needs to be identified.

Abstract Syntax:

The data syntax (machine independent) used to transfer each piece of information information needs to be defined.

Information Semantics:

Each piece of information needs to be described so that its meaning is clear. Each value transferred needs to be described, and the SI Units that are used for transfer also need to be described.

Transfer Protocol:

A method (format and rules) for transferring information.

Why is it Needed?

Descriptive Information Exchange becomes necessary when:

- Two Or More Data Bases Exist that Describe the Same Thing, and
- There is a need to transfer information from one data base to another.

The meaning and syntax of the data passed by the transfer mechanism must be understood by both the sender and the receiver.

What is the Approach?

1. Specify a common way of identifying and describing descriptive information.
2. Define a top level information model (set of related descriptive information) common to most systems.
3. Refine the information model for specific systems.
4. Integrate existing database definitions into the system specific information model.
5. Define the protocol for transferring information.

What are the Programmatic Problems?

1. Recognizing that there is a need.

2. Developing a structure for capturing an information model.
   - Capture the structure of the system.
     (e.g. Is X part of the system?)

   - Capture the data representation of each component of the system.
     (e.g. Is X a float or an integer?)

3. Disseminating the information model.

What are the Technical Problems?

Complexity of Model to Represent:

- Decomposition of System into Classes, Subclasses and Instances - Too Many Layers To Represent Reality

- Conversions for data type, format, units, and enumerated values

Part 2:
Architecture

Fundamental Concepts

Three Layer Entity-Relationship Model

Relationship Types are the top layer

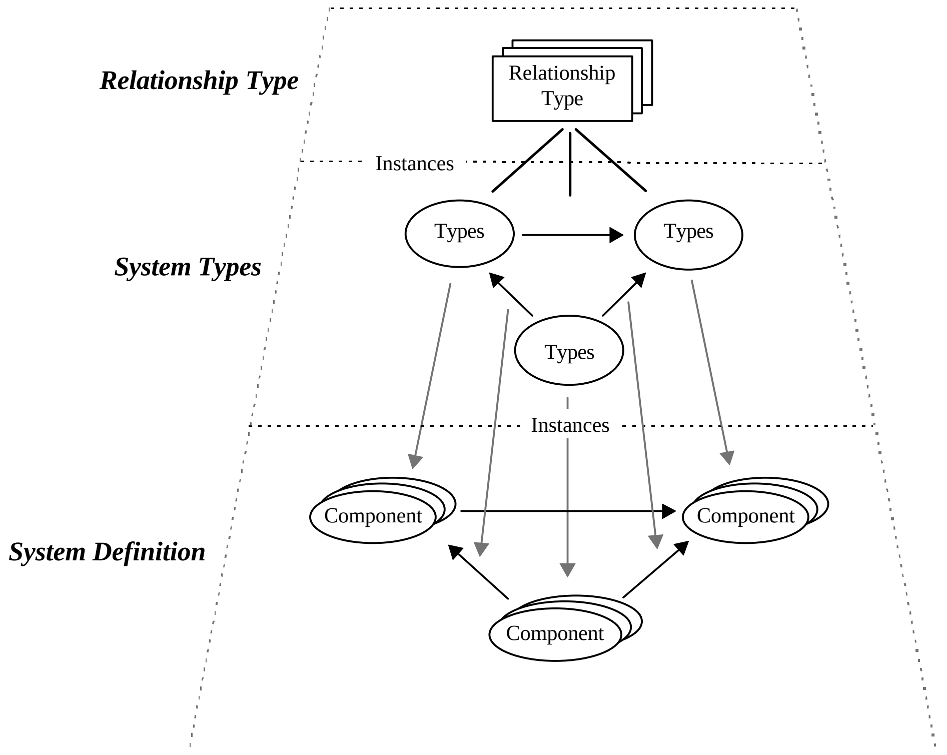
- Built-In Types and User Defined Types
- They define the semantics of the XDI Model

Relationships at the second layer define the Classes of the Info Model

- They are used to characterize entity types
- Each is an attribute of an entity type

Instances at Layer 3 Define the Entities of the Real System

Preliminary Architecture



Architectural Layers

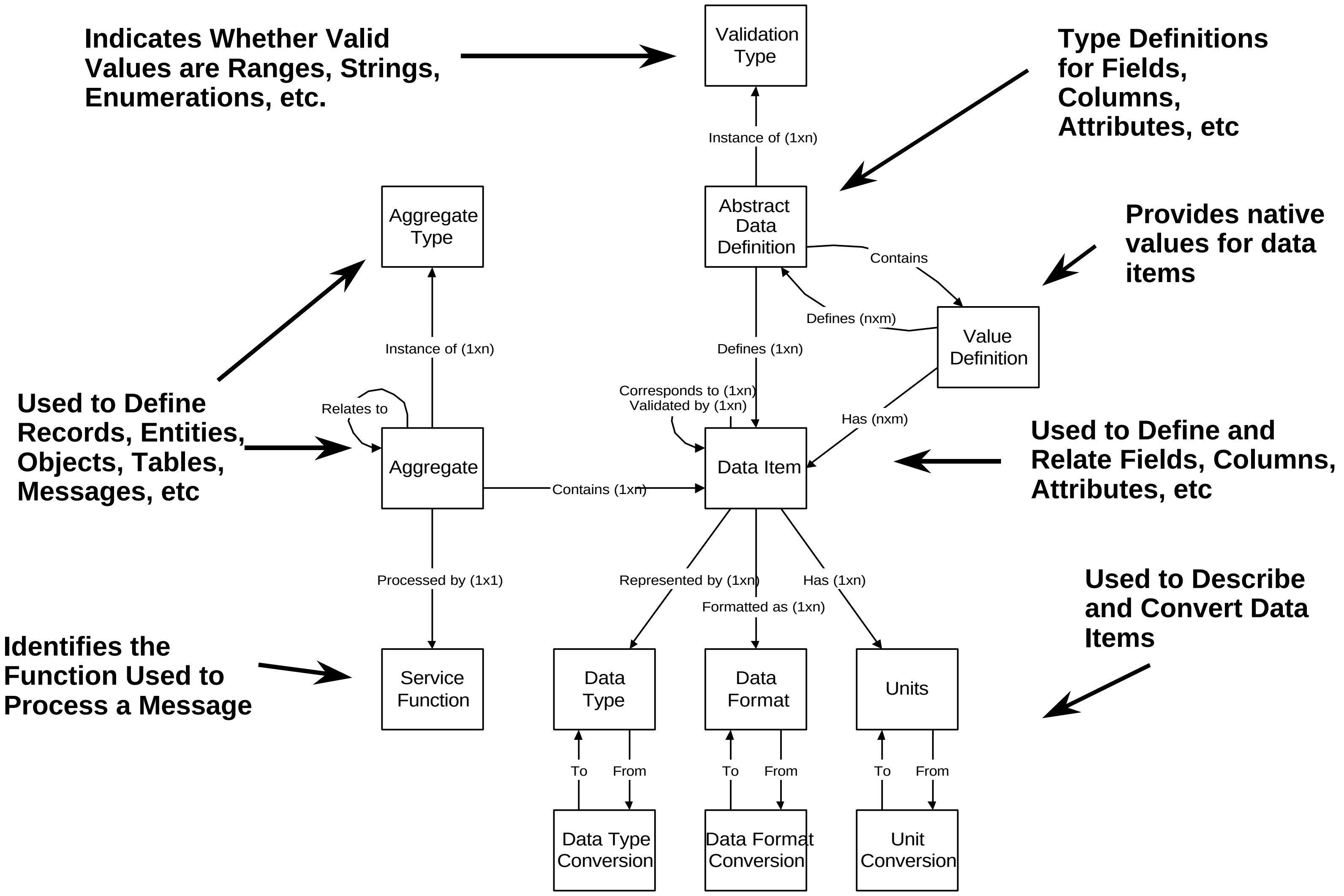
System Types	**Identifying, categorizing, and relating components. These represent the types of components that can be instantiated. (12 Port Mux).**
System Definition	**Instances of System Types. These define the components configured for the system.**

Data Representation Model

The Data Representation Model defines the semantics and syntax of the information to be transferred.

Only the identifiers of elements of the information model that are not data can be transferred.

Data Representation Model



Part 3:
Relationships

Builtin Relationships

Instantiation	Uses an element definition at one layer as a class definition for the layer below.
Inheritance	Used to indicate subclassing.
Data Definition	Used to define elements of a level as data.
Component	Used to define relationships between different types of system components

Component Relationships

Component Relationships have an attribute that indicates their presence in the instances (at the next lower level) of their source and target elements. This attribute is used to indicate whether a relationship is:

Mandatory	Indicates that this relationship is required for all instances of the source and target elements.
Optional	Indicates that this relationship is optional for instances of the source and target elements.
Conditional	Indicates that this relationship is conditional for instances of the source and target elements. When this value is selected, the constraint attribute is used to define the condition (using a Boolean expression).
SingleSelection	Indicates that this relationship is one of a list (of the same relationship type), of which only one may be selected for the source instance.
MultipleSelection	Indicates that this relationship is one of a list (of the same type), of which one or more may be selected.

Component Relationships have a pair of ordering attributes that are used to order (sequence) the System-Defined relationships for each of the source and target elements.

Part 4:
Status

Where Are We?

Second Use of the Model Has Been Implemented In a MS-Access DB

The System Model Has Been Simplified

The Data Model Portion Has Been Completed and Is Now Undergoing Validation

Issues

Most High Level Issues Have Now Been Resolved

Continued Testbedding Will Drive Out New Issues